Universal Adversarial Perturbations

Presenters:
Aleenah Khan
Kyle Rebello

About the paper

- **Authors:**
  - **Seyed-Mohsen Moosavi-Dezfooli**
  - **Alhussein Fawzi**
  - **Omar Fawzi**
  - **Pascal Frossard**
- **It was published in CVPR 2017.**
- **It has 1172 Citations.**
- **Link:**
  - "Deepfool: a simple and accurate method to fool deep neural networks,"

Outline

- Motivation
- Definition
- Contributions of the Paper
- Universal Perturbation
- Universal Perturbations for Deep Nets
  - Cross-model Universality
  - Visualization
  - Fine-tuning
- Conclusion
  - For
  - Against

Motivation

Can we find a *single* small image perturbation that fools a state-of-the-art deep neural network classifier on all natural images?

YES!

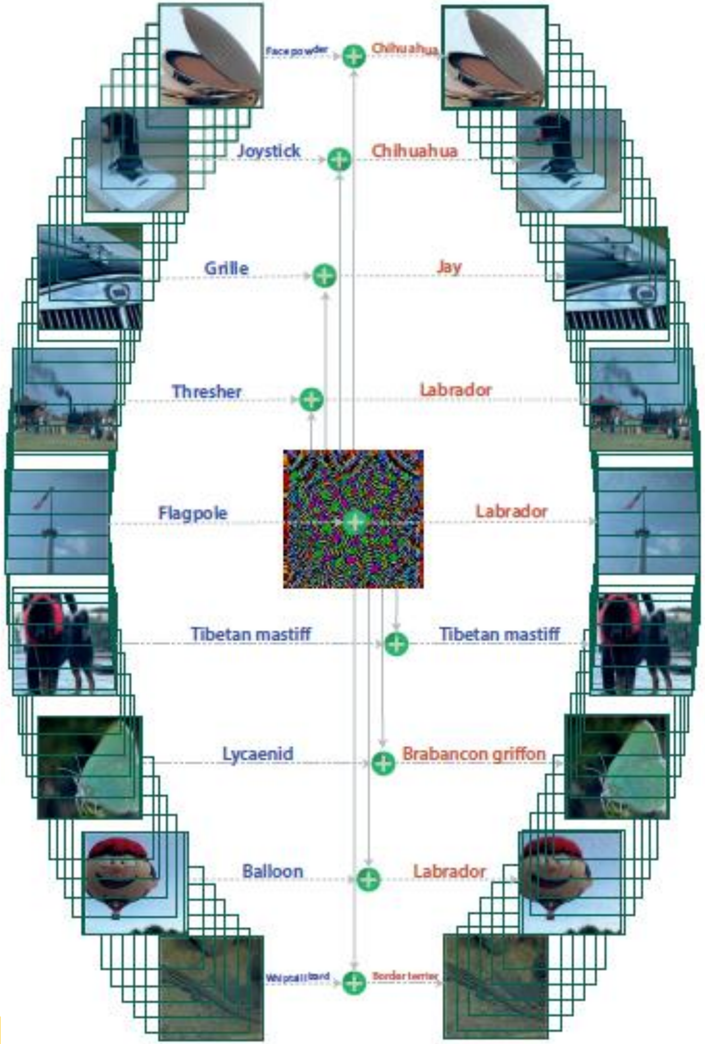
***Universal* perturbation vectors exist!**

Adding such a perturbation to natural images can fool the deep neural network to misclassify images with high probability.

Definition

These perturbations are:

- **Universal / Image-agnostic**
- **Quasi-imperceptible**



Contributions

- Existence of universal image-agnostic perturbations for state-of-the-art deep neural networks.
- Algorithm for finding universal perturbations.
- Proof for the generalization property across images.
- Proof for generalization across deep neural networks.

} Doubly Universal

- Analysis of the high vulnerability of deep neural networks to universal perturbations
  - Geometric correlation between different parts of the decision boundary.

UCF

Universal Perturbations

- Seek vector such that

$$\hat{k}(x + v) \neq \hat{k}(x) \text{ for "most" } x \sim \mu.$$

- μ = distribution of images
- $\hat{k}$ = classification function
- v = perturbed vector

Conditions

- **Vector should satisfy:**

$$1.\ \|v\|_p \le \xi,$$

$$2.\ \underset{x\sim\mu}{\mathbb{P}}\left(\hat{k}(x+v) \neq \hat{k}(x)\right) \ge 1-\delta.$$

- ξ **Controls magnitude of perturbation vector**
- δ **Quantifies desired fooling rate**

Algorithm

input: Data points X, classifier $\hat{k}$, desired ℓ_p norm of the perturbation ξ, desired accuracy on perturbed samples δ.
output: Universal perturbation vector v.
Initialize $v \leftarrow 0$.
while $\text{Err}(X_v) \leq 1 - \delta$ **do**
    for each datapoint $x_i \in X$ **do**
        if $\hat{k}(x_i + v) = \hat{k}(x_i)$ **then**
            Compute the *minimal* perturbation that sends $x_i + v$ to the decision boundary:

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$

            Update the perturbation:

$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i).$$

        end if
    end for
end while

UCF

Projecting Universal Perturbation

- **Projection operator**

$$\mathcal{P}_{p,\xi}(v) = \arg\min_{v'} \|v - v'\|_2 \text{ subject to } \|v'\|_p \leq \xi$$

- **Then update vector**

$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i)$$

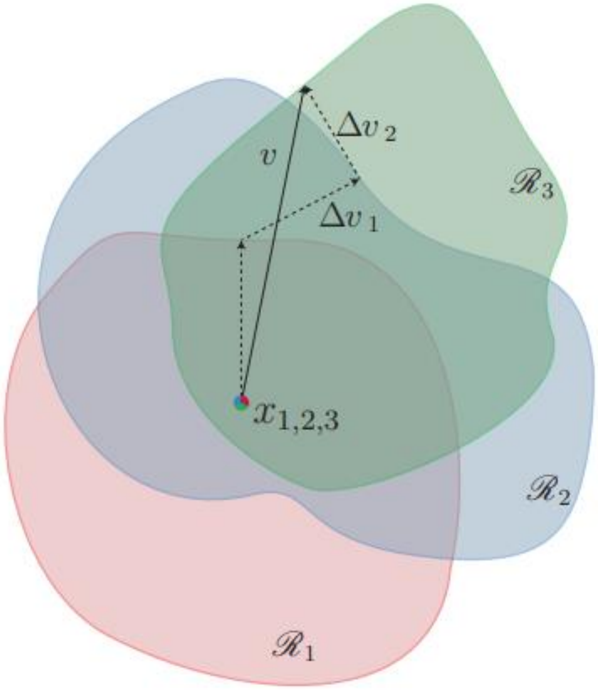
- **Perform iterations until**

$$\text{Err}(X_v) := \frac{1}{m} \sum_{i=1}^{m} 1_{\hat{k}(x_i+v) \neq \hat{k}(x_i)} \geq 1 - \delta$$

- **Where m is the number of datapoints to use from entire dataset**
- **m can be small and still compute an effective universal perturbation**

UCF

Universal Perturbation Visualization



- $\mathscr{R}_i$ = classification region
- Δv_i = minimal perturbation to move point outside of $\mathscr{R}_i$
- v = universal perturbation vector

Universal Perturbations for Deep Nets

- Experiment details:
  - Estimated universal perturbations for following neural networks:
    - CaffeNet, VGG-F, VGG-16, VGG-19, GoogLeNet, ResNet-152
  - ILSVRC 2012 validation set
    - 50,000 images
    - set X contains 10,000 images (i.e., in average 10 images per class)
  - Results are reported for:
    - $p = 2$ and $p = \infty$, where $\xi = 2000$ and $\xi = 10$ respectively.

Experimental Results

- Results reported on:
  - set X (used to compute the universal perturbation)
  - validation set (not used to compute the universal perturbation)

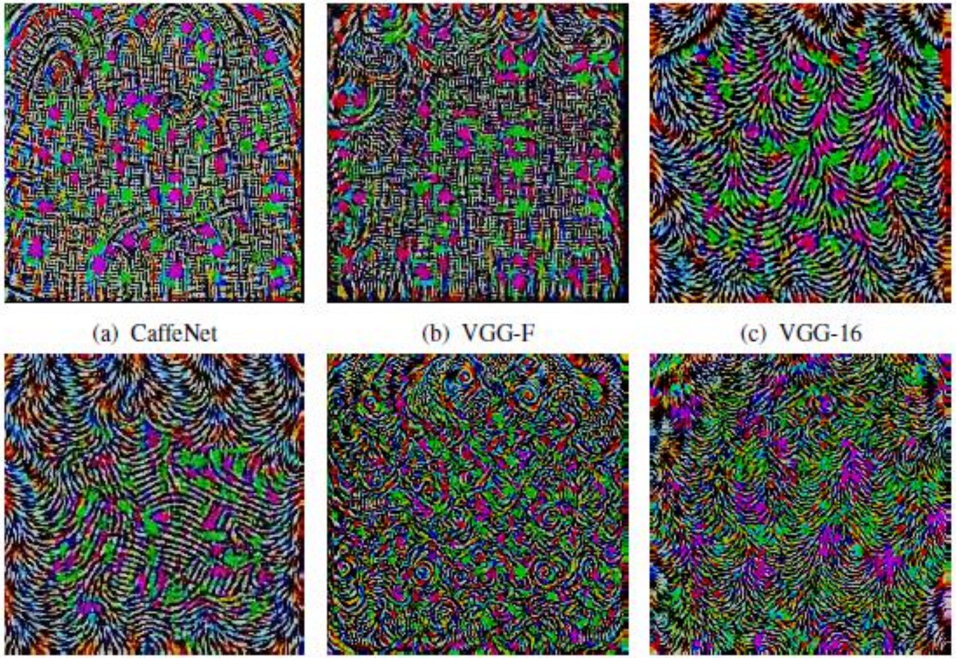
		CaffeNet [9]	VGG-F [3]	VGG-16 [18]	VGG-19 [18]	GoogLeNet [19]	ResNet-152 [7]
ℓ_2	X	85.4%	85.9%	90.7%	86.9%	82.9%	89.7%
	Val.	85.6%	87.0%	90.3%	84.5%	82.0%	88.5%
ℓ_∞	X	93.1%	93.8%	78.5%	77.8%	80.8%	85.4%
	Val.	93.3%	93.7%	78.3%	77.8%	78.9%	84.0%

Proof of Quasi-Imperceptibility

- **Visual examples using the GoogLeNet architecture**
- **Images belong to:**
  - **ILSVRC 2012 Validation Set**
  - **Mobile Phone Camera**

Visualization

- **Visualization of universal perturbations for different networks.**
- **These images are generated with $p = \infty$ and $\xi = 10$.**



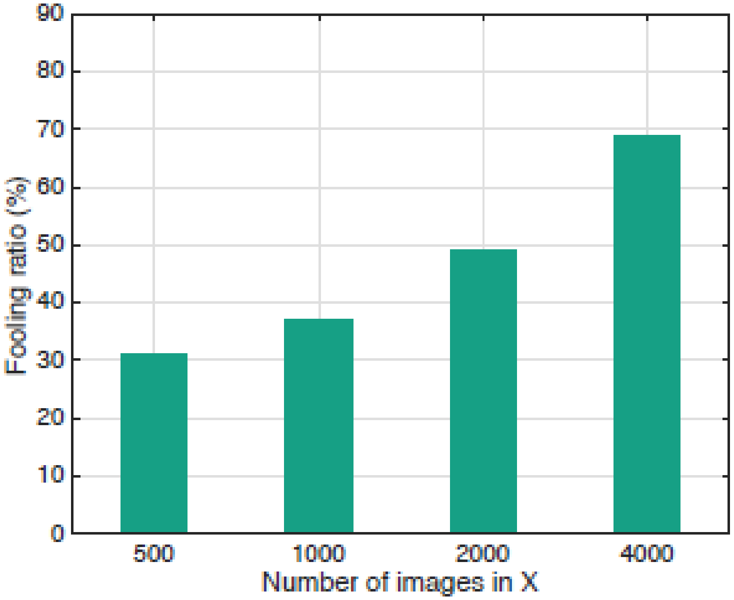
(a) CaffeNet (b) VGG-F (c) VGG-16 (d) VGG-19 (e) GoogLeNet (f) ResNet-152

Visualization

- **Universal perturbations are not unique.**
- **Diverse universal perturbations for the GoogLeNet architecture.**
- **Generated using different random shufflings of the set X.**
- **Normalized inner products for any pair of universal perturbations does not exceed 0.1.**

Effect of size of X on Quality

- If X = 500 images, more than 30% of the images can be fooled.
- This result is significant because the number of classes in ImageNet are 1000.
- A large set of unseen images can be fooled, even when set X contains less than one image per class!



Cross-Model Universality

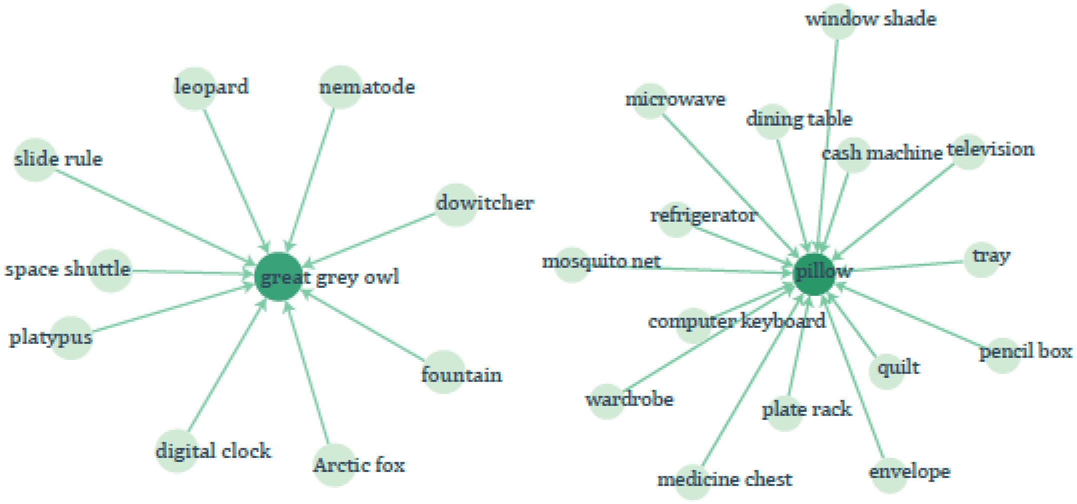
- **Universal perturbations computed for the VGG-19 network have a fooling ratio above 53% for all other tested architectures.**
- **For some architectures, the universal perturbations generalize very well across other architectures.**

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Visualization of the effect of Universal Perturbations

- A directed graph G = (V,E)
  - vertices = labels
  - directed edges e = (i → j) images of class i are fooled into label j

- Union of disjoint components.
- Connected Components.
- Existence of Dominant Labels.



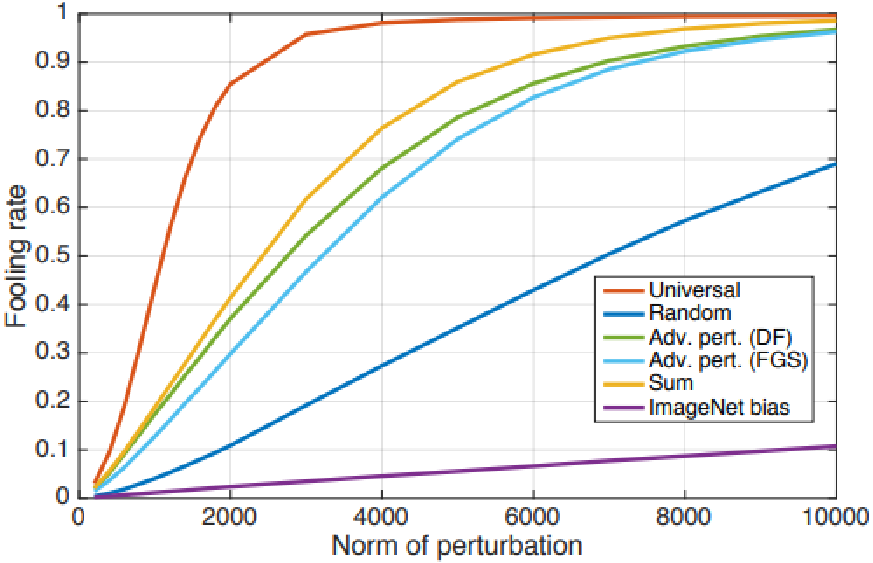
Fine-tuning with universal perturbations.

- Fine-tuned the VGG-F architecture by modifying training set.
- For each training point, a universal perturbation is added with probability 0.5.
- Pre-compute a pool of 10 different universal perturbations and picked randomly from this pool.
- Trained 5 extra epochs on the modified training set.

Attacking the Fine-tuned Network

- Computed a new universal perturbation for the fine-tuned network (with $p = \infty$ and $\xi = 10$).
- After 5 extra epochs, the fooling rate on the validation set is 76.2%.,
  - Originally it was 93.7%.
- Repeated the procedure
  - Obtained a new fooling ratio of 80.0%.
- The repetition of this procedure for a fixed number of times does not yield any improvement over 76.2%.
- Fine-tuning leads to a mild improvement in the robustness, it does not fully immune against universal perturbations.

Perturbation Comparison



- **Universal perturbation reaches high fooling rate quickly**

	Algorithm 1	Random Vectors
Fooling Rate	85%	10%

- **Suggests universal perturbation exploits geometric correlations of classifiers decision boundary**

UCF

Random vs Universal Perturbation

- Norm of random perturbation:

$$\Theta(\sqrt{d}\|r\|_2)$$

- d = dimension of input space
- $\|r\|_2$ = distance between data point and boundary
- For ImageNet classification task:

$$\sqrt{d}\|r\|_2 \approx 2 \times 10^4$$

- Where universal perturbation equals just 2000

Capturing Decision Boundary Geometry

- **For each image in validation set compute:**

$$r(x) = \arg\min_r \|r\|_2 \text{ s.t. } \hat{k}(x+r) \neq \hat{k}(x)$$

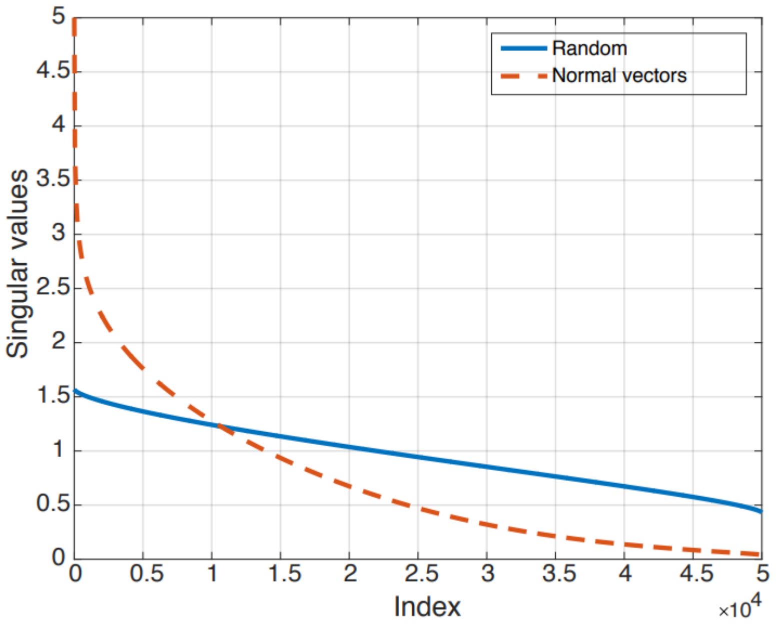
- **To quantify correlation between different regions:**

$$N = \left[\frac{r(x_1)}{\|r(x_1)\|_2} \cdots \frac{r(x_n)}{\|r(x_n)\|_2} \right]$$

UCF

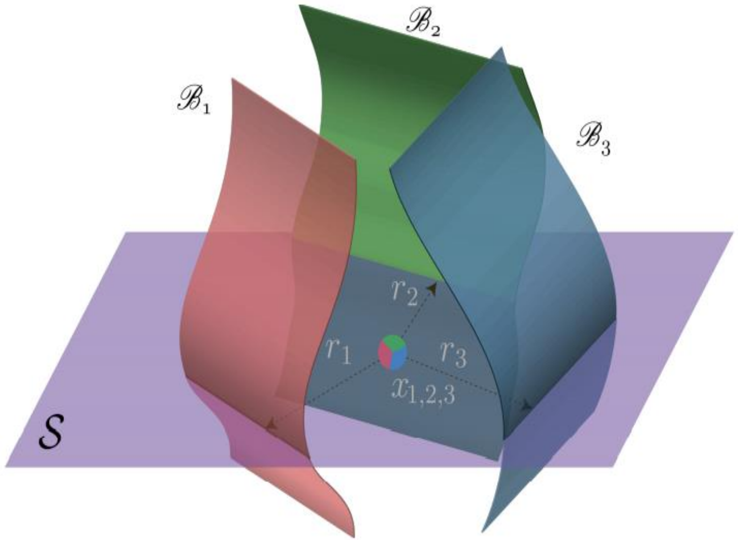
Capturing Decision Boundary Correlations

- The singular values of matrix N are computed
- The singular values of columns sampled uniformly and randomly from N are also computed
- Singular values from normal vectors decay quickly
- Singular values from random vectors decay slowly



Low Dimension Subspace Hypothesis

- $\mathcal{S}$ = low dimension subspace
- x_i = data point
- r_i = adversarial perturbation
- $\mathscr{B}_i$ = decision boundary



Low Dimension Subspace Verification

- Random vector of norm 2000 belonging to subspace S.
- Fooling ratio in well-sought subspace computed at 38%.
- Compared to 10% when doing random perturbations.
- This also helps explain why the universal perturbation generalizes well.

Conclusion

- **Showed the existence of small universal perturbations that can fool state-of-the-art classifiers on natural images.**
- **Proposed an iterative algorithm to generate universal perturbations.**
- **Highlighted several properties of universal perturbations.**
  - **Image-agnostic**
  - **Network-agnostic**
- **Explained the existence of universal perturbations with the correlation between different regions of the decision boundary.**
- **Provided insights on the geometry of the decision boundaries of deep neural networks.**

For

- This algorithm is able to generate a universal perturbation with a small sample of the data.
- Finding the subspace that allows the universal perturbation to be so effective.
- Finding geometric correlations between different parts of the decision boundary.
- The universal perturbation is image-agnostic and network-agnostic.

Against

- Used only a single dataset of natural images ImageNet for all experiments.
- The proposed method is expensive as it’s iterative.
- Performed fine-tuning on just a single architecture VGG-F.
- Fine-tuning procedure helped improve the fooling rate to 76.2% only.
- Their hypothesis for dominant labels need to be investigated.

Thank You